

Claims

1. In a communications network, a system for enabling both a host computer and a remote computer to view and control a software application controlling a hardware or software device running on the host computer, said system comprising:

a) a host computer running a software application, said software application having:

i) a core component controlling both the operation of the software application and a device attached to the computer;

ii) a graphical user interface presenting output from and accepting input to the software application;

iii) a communication module handling messages passed between the core component and the graphical user interface;

iv) a client/server module having a socket allowing bi-directional communications with other instances of said software application running on other computers in a network; and

v) a thread to monitor whether the core component has responded to a wait command; and

b) a remote computer running another instance of the software application, said software application connected to the host computer's software application by a TCP/IP connection;

wherein the host computer's software application and the remote computer's software application share information and control of both the software application running on the host computer and the hardware or software device connected to the host computer.

2. The system of claim 1 wherein the device is a hardware device.

3. The system of claim 2 wherein the hardware device is a microprocessor.

4. The system of claim 2 wherein the hardware device is an in-circuit emulator.

5. The system of claim 2 wherein the hardware device is a data storage device.

6. The system of claim 2 wherein the hardware device is a test instrument.

7. The system of claim 2 wherein the hardware device is an internet-enabled appliance.

8. The system of claim 1 wherein the device is a software device.

9. The system of claim 8 wherein the software device is a simulator.

10. The system of claim 1 wherein the host computer software application is configured as server-side software and the remote computer software application is configured as client-side software.

11. The system of claim 1 wherein the host computer software application is configured as client-side software and the remote computer software application is configured as server-side software.

12. The system of claim 1 wherein the TCP/IP connection is initiated by the software application configured as client-side software.

13. The system of claim 1 wherein a user of the remote computer is providing customer support to a user of the host computer.

14. The system of claim 1 wherein a user of the host computer is providing customer support to a user of the remote computer.

15. The system of claim 1 wherein a user of the remote computer is collaborating on a project with a user of the host computer.

16. The system of claim 1 wherein the computer running the software application which has been configured as client-side software is located behind a firewall or a similar Internet security device.

17. The system of claim 1 wherein the software application further includes a plurality of threads to pass messages to different components and modules of the software application.

18. The system of claim 17 wherein one of the threads passes messages from a queue in the communication module to the graphical user interface.

19. The system of claim 17 wherein one of the threads passes messages from the core component to a queue in the communication module.

20. The system of claim 17 wherein one of the threads takes messages from the socket and passes them to a queue in the communication module.

21. The system of claim 17 wherein one of the threads takes messages from the socket and passes them to the host software application's core component.

22. The system of claim 1 wherein the software application possesses a message-driven architecture.

23. The system of claim 1 wherein the messages passed by the communication module from the graphical user interface to the core component are commands inputted by one of the users.

24. The system of claim 1 wherein the messages passed by the communication module from the graphical user interface to the core component are requests from one of the users for information about the device.

25. The system of claim 1 wherein the messages passed by the communication module from the core component to the graphical user interfaces concern events that have occurred to the device.

26. The system of claim 1 wherein the messages passed by the communication module from the core component to the graphical user interfaces are responses to the users' request for information.

27. In a communications network, a method for enabling both a host computer and a remote computer to view and control a hardware or software device connected to the host computer and a software application controlling a hardware or software device running on the host computer, said method comprising:

- a) connecting a device to a host computer;
- b) running the same software application on both a remote computer and the host computer, said software application capable of controlling the device attached to the host computer and having:
 - i) a core component controlling both the operation of the software application and a device attached to the computer;
 - ii) a graphical user interface presenting output from and accepting input to the software application;
 - iii) a communication module handling messages passed between the core component and the graphical user interface; and
 - iv) a client/server module having a socket allowing bi-directional communications with other instances of said software application running other on other computers in a network; and
 - v) a thread to monitor whether the core component has responded to a wait command;
- c) establishing a TCP/IP connection between the remote software application and the host software application;
- d) transmitting commands and events specific to the software application and/or the device over said connection;

e) sharing graphical user interface information between the remote software application and the host software application such that a remote user and host user may provide input to and view output from the software application and the device connected to the host computer which is controlled by said software application running on the host computer; and

f) closing said TCP/IP connection between the remote computer and the host computer at the end of each session.

28. The method of claim 27 wherein the remote user is providing customer support to the host user.

29. The method of claim 27 wherein the host user is providing customer support to the remote user.

30. The method of claim 27 wherein the remote user and the host user are collaborating on a project.

31. The method of claim 27 wherein the software application configured as client-side software initiates the TCP/IP connection with the software application configured as server-side software.

32. The method of claim 27 wherein the software application configured as client-side software shuts down the TCP/IP connection with the software application configured as server-side software.

33. The method of claim 27 wherein the software application configured as server-side software shuts down the TCP/IP connection with the software application configured as client-side software.

34. In a communications network, a system for allowing both a host computer and remote computer to view and control a software application running on the host computer, said system comprising:

a) a host computer running a software application, said software application having:

i) a core component controlling the operation of the software application;

ii) a graphical user interface presenting output from and accepting input to the software application;

iii) a communication module handling messages passed between the core component and the graphical user interface;

iv) a client/server module having a socket allowing bi-directional communications with other instances of said software application running on other computers in a network; and

v) a thread to monitor whether the core component has responded to a wait command; and

b) a remote computer running another instance of the software application, said software application connected to the host computer's software application by a TCP/IP connection;

wherein the host computer and the remote computer share the graphical user interface providing information and control of the software application running on the host computer.

35. The system of claim 34 wherein the host computer software application is configured as client-side software and the remote computer software application is configured as server-side software.

36. The system of claim 34 wherein the host computer software application is configured as server-side software and the remote computer software application is configured as client software.

37. The system of claim 34 wherein the host computer is located behind a firewall or a similar network security device.

38. The system of claim 34 wherein a user of the remote computer is providing customer support to a user of the host computer.

39. The system of claim 34 wherein a user of the host computer is providing customer support to a user of the remote computer.

40. The system of claim 34 wherein a user of the remote computer is collaborating on a project with a user of the host computer.

41. The system of claim 34 wherein the messages passed by the communication module from the graphical user interface to the core component are commands inputted by one of the users.

42. The system of claim 34 wherein the software application further includes a plurality of threads to pass messages to different components and modules of the software application.

43. The system of claim 42 wherein one of the threads passes messages from a queue in the communication module to the graphical user interface.

44. The system of claim 42 wherein one of the threads passes messages from the core component to a queue in the communication module.

45. The system of claim 42 wherein one of the threads takes messages from the socket and passes them to a queue in the communication module.

46. The system of claim 42 wherein one of the threads takes messages from the socket and passes them to the host software application's core component.

47. The system of claim 34 wherein the software application possesses a message-driven architecture.

48. In a communications network, a method for allowing both a host computer and a remote computer to view and control a software application on the host computer, said method comprising:

a) running the same software application on both a remote computer and a host computer, said software application having:

i) a core component controlling the operation of the software application;

ii) a graphical user interface presenting output from and accepting input to the software application;

iii) a communication module handling messages passed between the core component and the graphical user interface; and

iv) a client/server module having a socket allowing bi-directional communications with other computers in a network running other instances of said software application; and

v) a thread to monitor whether the core component has responded to a wait command; and

b) establishing a TCP/IP connection between the remote software application and the host software application;

c) transmitting commands and events specific to the software application and/or the device over said connection;

d) sharing graphical user interface information between the remote software application and the host software application such that a remote user and a host user may provide input to and view output from the software application running on the host computer;

e) shutting down said TCP/IP connection between the remote computer and the host computer at the end of each session.

49. The method of claim 48 wherein the remote user is providing customer support to the host user.

50. The method of claim 48 wherein the host user is providing customer support to the remote user.

51. The method of claim 48 wherein the remote user and the host user are collaborating on a project.

52. The method of claim 48 wherein the software application configured as client-side software initiates the TCP/IP connection with the software application configured as server-side software.

53. The method of claim 48 wherein the software application configured as client-side software shuts down the TCP/IP connection with the software application configured as server-side software.

54. The method of claim 48 wherein the software application configured as server-side software shuts down the TCP/IP connection with the software application configured as client-side software.

55. In a communications network, a system for enabling a plurality of software applications to remotely control and monitor a software application running on a host computer, said system comprising:

- a) a plurality of computers; and
 - b) a plurality of identical software applications running on each of said computers, each instance of said software application having:
 - i) a core component controlling the operation of the software application;
 - ii) a graphical user interface presenting output from and accepting input to the software application;
 - iii) a communication module handling messages passed between the core component and the graphical user interface; and
 - iv) a client/server module having a socket allowing bi-directional communications with other instances of said software application running on other computers in a network; and
 - v) a thread to monitor whether the core component has responded to a wait command;
- each software application connected to the other software applications via a TCP/IP connection;
- wherein each software application can view and control the software application running on a chosen computer.

56. The system of claim 55 wherein one of the computers has a device attached which is controlled by the software application.

57. The system of claim 55 wherein the other connected software applications can view and control the attached device.

58. The system of claim 56 wherein the device is a hardware device.

59. The system of claim 58 wherein the hardware device is an in-circuit emulator.

60. The system of claim 58 wherein the hardware device is a data storage device.

61. The system of claim 58 wherein the hardware device is a test instrument.

62. The system of claim 58 wherein the hardware device is an internet-enabled appliance.

63. The system of claim 58 wherein the hardware device is a microprocessor.

64. The system of claim 56 wherein the device is a software device.

65. The system of claim 64 wherein the software device is a simulator.

66. The system of claim 55 wherein users of a plurality of remote computers are providing customer support to a user of a host computer.

67. The system of claim 55 wherein the user of the host computer is providing customer support to a user of a remote computer.

68. The system of claim 55 wherein users of a plurality of remote computers are collaborating on a project with a user of a host computer.

69. The system of claim 55 wherein the host computer is located behind a firewall or a similar network security device.

70. The system of claim 55 wherein the computer running the software application which has been configured as client-side software is located behind a firewall or a similar Internet security device.

71. The system of claim 55 wherein the messages passed by the communication module from the graphical user interface to the core component are commands inputted by one of the users.

72. The system of claim 55 wherein the messages passed by the communication module from the graphical user interface to the core component are requests from one of the users for information about the device.

73. The system of claim 55 wherein the messages passed by the communication module from the core component to the graphical user interface concern events that have occurred to the device.

74. The system of claim 55 wherein the messages passed by the communication module from the core component to the graphical user interface are responses to the users' request for information.

75. The system of claim 55 wherein the software application further includes a plurality of threads to pass messages to different components and modules of the software application.

76. The system of claim 75 wherein one of the threads passes messages from a queue in the communication module to the graphical user interface.

77. The system of claim 75 wherein one of the threads passes messages from the core component to a queue in the communication module.

78. The system of claim 75 wherein one of the threads takes messages from the socket and passes them to a queue in the communication module.

79. The system of claim 75 wherein one of the threads takes messages from the socket and passes them to the host software application's core component.

80. The system of claim 55 wherein one of the software applications is configured as a server-side software.

81. The system of claim 80 wherein the remainder of the software applications are configured as client-side software.

82. The system of claim 55 wherein the software application possesses a message-driven architecture.

NOH-001.CLS